

Model symulacji — raport częściowy — Projekt MOPS

Błażej Sewera, Mateusz Winnicki, Wojciech Kowalski

1 grudnia 2020

Projekt MOPS

Cele projektu

Celem projektu jest zasymulowanie systemu dwóch węzłów sieciowych, obsługujących n źródeł ruchu typu ON-OFF, zdolnych generować pakiety o stałej długości L w stanie ON.

Każdy węzeł sieciowy modelujemy jako nieskończoną kolejkę FIFO, każdy ma podłączoną dowolną liczbę źródeł ruchu.

Wartości teoretyczne

Czas trwania stanu ON jest opisywany rozkładem wykładniczym:

$$t_{ON} = \lambda_{ON} e^{-\lambda_{ON} t}$$

Średni czas trwania stanu ON wynosi:

$$\mathbb{E}t_{ON} = \frac{1}{\lambda_{ON}}$$

Pakiety będą generowane przez cały czas trwania stanu ON.

Czas trwania stanu OFF również jest opisywany rozkładem wykładniczym:

$$t_{OFF} = \lambda_{OFF} e^{-\lambda_{OFF} t}$$

Średni czas trwania stanu OFF wynosi:

$$\mathbb{E}t_{OFF} = \frac{1}{\lambda_{OFF}}$$

Podczas tego stanu pakiety nie będą generowane.

Moduły modelu symulacyjnego

- Program główny — inicjalizuje moduły i nadzoruje ich pracę
- Zegar symulacji — reprezentuje czas symulacji, od niego zależą zdarzenia
- Algorytm zdarzeniowy — uaktualnia stan systemu i liczniki metryk
- Stan systemu — zbiór parametrów symulacji
- Algorytm czasowy — informuje system o obecnie trwającym stanie, kontroluje czas trwania stanu (wygenerowany przez algorytm zmiennych losowych) oraz wywołuje zdarzenia (a w konsekwencji algorytm zdarzeniowy)
- Algorytm zmiennych losowych — generuje czas trwania stanów ON i OFF
- Generator raportów — odpowiedzialny za zebranie i prezentację metryk

Badane metryki pomiarowe

- Średnia liczba pakietów w kolejce l_{queue} ,
- średni czas oczekiwania w kolejce t_{wait} ,
- średnie opóźnienie przekazu pakietu, definiowane jako średnia suma czasu, oczekiwania w kolejce oraz czasu odebrania całego pakietu przez węzeł sieciowy. Jako że długość pakietu jest stała i wynosi L , czas odebrania pakietu przez węzeł również będzie stały $t_{delay} = t_{wait} + T_{receive}$,
- średnie obciążenie serwera.

Przedstawiony model zaimplementujemy w języku programowania Python.

Scenariusz

System dwóch węzłów obsługujących do n źródeł ruchu. Pierwszy węzeł obsługuje N źródeł ruchu mających te same parametry (oznaczone jako $T1$). Ruch z M źródeł jest przekazywany do węzła drugiego (oznaczone jako P), natomiast ruch z pozostałych źródeł, ponieważ idzie do innych (nieistniejących w symulacji) węzłów, nie jest przekazywany dalej. Do węzła drugiego, oprócz ruchu z węzła pierwszego dociera jeszcze dodatkowy ruch z K źródeł. N , M oraz K są parametrami wejściowymi symulacji.

Uproszczenia przyjęte w projekcie

- Nie tworzymy nowego procesu dla każdego klienta, ponieważ nie mamy na celu symulować realnej komunikacji klient-serwer; zamiast tego będziemy agregować teoretyczną ilość danych wysyłanych przez klienty znajdujące się w stanie ON.
- Z racji tego, że zakładana kolejka K jest nieskończona, za stracone uznawać powinniśmy tylko pakiety, które nie zostały odebrane. W projekcie nie symulujemy jednak odbierania pakietów z określonym prawdopodobieństwem, więc poziom strat pakietów, definiowany jako stosunek różnicy pakietów wysłanych i odebranych do liczby pakietów wysłanych byłby zawsze równy zero.

Źródła

- Laboratorium sieci usługowych — pomiary w sieciach IP http://aai.tele.pw.edu.pl/data/SWUS/swus_lab_pomiary.pdf
- Fragmenty materiałów wykładowych z przedmiotu Rachunek Prawdopodobieństwa PWR <http://prac.im.pwr.wroc.pl/~wkosz/RP2012.pdf>
- Wikipedia — strony poświęcone m.in. Teorii kolejek i stronach pochodnych https://en.wikipedia.org/wiki/Queueing_theory
- Wykład MOPS, ze szczególnym uwzględnieniem tematów 5 i 6
- Wykłady dr inż. Joanny Matysiak z przedmiotu Probabilistyka <http://mini.pw.edu.pl/~jmatysiak/>

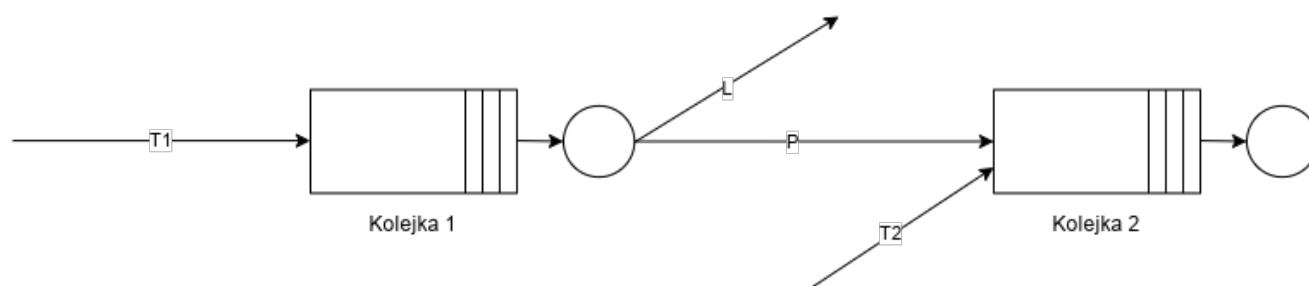
Załączniki

T1 - n źródeł ruchu generujących pakiety wchodzące do kolejki 1.

T2 - do n źródeł ruchu generujących pakiety wchodzące do kolejki 2.

P - pakiety przechodzące z pierwszego węzła do drugiego

L - pakiety opuszczające węzeł pierwszy, ale nie wchodzące do węzła 2.



Rysunek 1: Scenariusz